

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Claim 1 (Previously Presented): A method implemented in a computer, the method comprising:

generating an array update operation based on a query to update a relational database;

wherein said array update operation specifies a plurality of row-identifier and value pairs to update multiple rows in a table of said relational database;

repeatedly finding, and storing in a structure, a block-identifier of a block that contains a row of data identified by a row-identifier in at least a group of row-identifier and value pairs in said plurality, by use of a database index prior to retrieval of the block;

wherein said structure is located in a memory of said computer;

wherein each value comprises data to be updated in said row identified by said row-identifier;

performing a single access operation without context switching, to retrieve from a storage device and to store in a cache, a plurality of blocks of data of said table, said blocks being identified by a corresponding plurality of block-identifiers in the structure;

wherein several of said blocks are non-contiguous in said storage device; and

repeatedly updating, in blocks in the cache, each row identified in the group of row-identifier and value pairs, using a corresponding value in the row-identifier and value pairs; and

sorting the block-identifiers, prior to retrieval of the blocks by performing the single access operation.

Claim 2 (canceled).

Claim 3 (Previously Presented): The method of claim 1 wherein:

the sorting is performed subsequent to storage of the block-identifiers in the structure.

Claim 4 (Previously Presented): The method of claim 1 further comprising:

subsequent to said finding and prior to said storing, checking if the block-identifier has a duplicate already stored in the structure and if so then not storing the block-identifier in the structure.

Claim 5 (Previously Presented): The method of claim 1 further comprising, prior to updating:

repeating said finding of block-identifiers for all row-identifiers in the group of row-identifier and value pairs.

Claim 6 (Previously Presented): The method of claim 1 wherein:

the table is organized in a hash cluster.

Claim 7 (Previously Presented): The method of claim 1 wherein:

the database index is a B-tree index.

Claim 8 (Previously Presented): The method of claim 1 wherein:

said structure comprises an array; and

the array has a plurality of entries identical in number to the plurality of blocks held in the cache.

Claim 9 (Previously Presented): The method of claim 1 further comprising:

writing a plurality of logs, at least one log for each row identified in the group of row-identifier and value pairs.

Claim 10 (Previously Presented): The one or more storage media of claim 13 further comprising:

instructions to unpin each block after updating all rows in said each block; and

instructions to flush an unpinned block to said storage device only when another block needs space in the cache occupied by the unpinned block.

Claim 11 (Currently Amended): The one or more storage media of claim 13 wherein:

a plurality of file offsets are provided to the single access operation, one file offset for each block in the group;

wherein each file offset is an offset in a file from where reading of data is to begin.

Claim 12 (canceled).

Claim 13 (Currently Amended): One or more storage media in which are stored instructions to perform a computer-implemented method, the instructions comprising:

instructions to generate an array update operation based on a query to update a relational database;

wherein said array update operation specifies a plurality of row-identifier and value pairs to update multiple rows in a table of said relational database;

instructions to repeatedly find and store in a structure, a block-identifier of a block that contains a row identified by a row-identifier in at least a group of row-identifier and value pairs in said plurality, by use of a database index of said relational database;

instructions to invoke performance of a single operation without context switching during said performance, to retrieve from a storage device and to store in a cache, a plurality of blocks, said blocks being identified by block-identifiers in the structure;

instructions to repeatedly update in blocks in the cache, each row identified in the group of row-identifier and value pairs, using a corresponding value in the row-identifier and value pairs;

wherein several of said blocks are non-contiguous in said storage device; and

instructions to sort the block-identifiers, prior to retrieval of the blocks by performing the single access operation.

Claim 14 (Previously Presented): The one or more storage media of claim 13 further comprising said structure storing the block-identifiers.

Claim 15 (Currently Amended): A computer comprising a processor and one or more storage media coupled to the processor, the one or more storage media comprising:

instructions to automatically generate an array update operation based on a query to update a relational database;

instructions to automatically use a database index to look up a block-identifier of a block that contains a row identified by an identifier in a plurality of identifier and value pairs to be used to perform said array update operation on a table in said relational database;

instructions to automatically store the block-identifier in a structure in memory;

instructions to automatically repeat said instructions to automatically use and said instructions to automatically store, for all identifiers in at least a group of identifier and value pairs in said plurality;

instructions to automatically perform a vector read operation, to retrieve from a disk and store in a cache, each block in a group of blocks identified by block-identifiers stored in said structure, wherein the group of blocks are all stored in the cache during execution of a single function call;

wherein several of said blocks in the group are non-contiguous in said disk;

instructions to automatically modify a row in a block stored in the cache, using a value in the plurality of identifier and value pairs;

instructions to automatically repeat said instructions to automatically modify, with each row identified in the group of identifier and value pairs; and

instructions to sort said block-identifiers prior to retrieval of the blocks from the storage device by performing the ~~single-access~~ vector read operation.

Claim 16 (Currently Amended): An apparatus comprising a database, the apparatus comprising:

means for generating an array update operation based on a query to update the database;

wherein said array update operation specifies a plurality of row-identifier and value pairs to update multiple rows in a table of the database;

means for using a database index to look up a block-identifier of a block that contains the row identified by an identifier in the plurality of identifier and value pairs;

means for storing the block-identifier in a structure in memory;

means for repeating (using the database index to look up and storing the block-identifier), for all identifiers in at least a group of identifier and value pairs;

means for invoking performance of a vector read operation without context switching, to retrieve from a disk and store in a cache, each block in a group of blocks identified by block-identifiers stored in said structure, wherein the group of blocks are all stored in the cache during execution of a single function call;

wherein several of said blocks in the group are non-contiguous in said disk;

means for modifying a row in a block stored in the cache, using a value in the plurality of identifier and value pairs; and

means for repeating said modifying with each row identified in the group of identifier and value pairs;

wherein said block-identifiers are sorted prior to retrieval of the blocks from the storage device by performing the ~~single-access~~ vector read operation.

Claim 17 (Currently Amended): One or more storage media comprising instructions to perform a method implemented in a computer, the one or more storage media comprising:

instructions to generate an array update operation based on a query to update a database;

wherein said array update operation specifies a plurality of row-identifier and value pairs to update multiple rows in a table of said database;

instructions to find a block-identifier of a block that contains the row identified by a row-identifier in a row-identifier and value pair in said plurality, by use of a database index in said database;

instructions to store the block-identifier in a structure in memory;

instructions to repeatedly (find the block-identifier and store the block-identifier), for all row-identifiers in at least a group of row-identifier and value pairs in said plurality;

instructions to invoke performance of a vector read operation without context switching during said performance, to retrieve from a storage device and store in a cache, each block in a group of blocks identified by block-identifiers stored in said

structure, wherein the group of blocks are all stored in the cache during execution of a single function call;

wherein several of said blocks in the group are non-contiguous in said storage device;

instructions to update the row in the block in the cache, using the value in the row-identifier and value pair;

instructions to repeatedly execute said instructions to update, with each row identified in the group of row-identifier and value pairs; and

instructions to sort said block-identifiers prior to retrieval of the blocks from the storage device by performing the ~~single access~~ vector read operation.

Claim 18 (Previously Presented): The one or more storage media of claim 13 comprising at least one of a disk, a chip and a cartridge.

Claim 19 (Currently Amended): The one or more storage media of claim 13 wherein:

the blocks are sorted during execution of said instructions to sort based on adjacency such that during performance of said single ~~access~~ operation, block-identifiers of blocks physically adjacent to one another at a periphery of a disk in the storage device are presented at one time to the storage device and identifiers of blocks that are physically adjacent to one another and located closer to a center of the disk are presented at another time.

Claim 20 (Previously Presented): The computer of claim 15 wherein:

the blocks are sorted during execution of said instructions to sort based on adjacency such that block-identifiers of blocks physically adjacent to one another at a periphery of said disk are presented at one time to a disk drive comprising said disk and identifiers of blocks that are physically adjacent to one another and located closer to a center of said disk are presented at another time.

Claim 21 (Previously Presented): The one or more storage media of claim 13 further comprising:

instructions to perform a write operation from said cache to said storage device when space is needed in said cache.

Claim 22 (Previously Presented): The one or more storage media of claim 13 further comprising:

instructions to store a row of said table in a hash cluster.

Claim 23 (Previously Presented): The one or more storage media of claim 13 wherein said storage device comprises at least one of a disk, a chip and a cartridge.

Claim 24 (Previously Presented): The one or more storage media of claim 17 further comprising:

instructions to store a row of said table in a hash cluster.

Claim 25 (Previously Presented): The one or more storage media of claim 17 wherein said storage device comprises at least one of a disk, a chip and a cartridge.